2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Constrained Real-Parameter Optimization with Generalized Differential Evolution

Saku Kukkonen and Jouni Lampinen

Abstract— **This paper presents results for the CEC 2006 Special Session on Constrained Real-Parameter Optimization where the Generalized Differential Evolution (GDE) has been used to solve given test problems. The given problems consist of 24 problems having one objective function and one or more in-/equality constraints. Almost all the problems were solvable in a given maximum number of solution candidate evaluations. The paper also shows how GDE actually needs lower number of function evaluations than usually required.**

I. Introduction

Many practical problems have multiple objectives and several aspects cause multiple constraints in the problems. For example, mechanical design problems have several objectives such as obtained performance and manufacturing costs, and available resources may cause limitations. Constraints can be divided into boundary constraints and constraint functions. Boundary constraints are used when the value of a decision variable is limited to some range, and constraint functions represent more complicated constraints, which are expressed as functions.

A mathematically constrained multi-objective optimization can be presented in the form:

$$\begin{array}{ll} \text{minimize} & \{f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})\} \\ \text{subject to} & (g_1(\vec{x}), g_2(\vec{x}), \ldots, g_K(\vec{x}))^T \leq \vec{0}. \end{array}$$

Thus, there are M functions to be optimized and K constraint functions. Maximization problems can be easily transformed to minimization problems and different constraints can be converted into form $g_j(\vec{x}) \leq 0$, thereby the formulation above is without loss of generality.

Problems used in this paper are single-objective problems with constraints, and they have been defined in [1] for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. The problems have different number of variables and in-/equality constraints, and difficulties of functions vary from linear to non-linear. Also evaluation criteria are given in [1].

This paper continues describing the basic Differential Evolution (DE) in Section II and its extension Generalized Differential Evolution (GDE) in Section III. Section IV describes experiments and finally conclusions are given in Section V.

The authors are with the Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland; email: saku.kukkonen@lut.fi.

II. Differential Evolution

The DE algorithm [2], [3] was introduced by Storn and Price in 1995 and it belongs to the family of Evolutionary Algorithms (EAs). The design principles of DE are simplicity, efficiency, and the use of floating-point encoding instead of binary numbers. As a typical EA, DE has a random initial population that is then improved using selection, mutation, and crossover operations. Several ways exist to determine a stopping criterion for EAs but usually a predefined upper limit G_{max} for the number of generations to be computed provides an appropriate stopping condition. Other control parameters for DE are the crossover control parameter CR, the mutation factor F, and the population size NP.

In each generation G, DE goes through each D dimensional decision vector $\vec{x}_{i,G}$ of the population and creates the corresponding trial vector $\vec{u}_{i,G}$ as follows in the most common DE version, DE/rand/1/bin [4]:

$$\begin{array}{l} r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}, \text{(randomly selected,} \\ \quad \text{except mutually different and different from } i) \\ j_{rand} = \text{floor}\,(rand_i[0,1) \cdot D) + 1 \\ \text{for}(j = 1; j \leq D; j = j+1) \\ \{ \\ \quad \text{if}(rand_j[0,1) < CR \vee j = j_{rand}) \\ \quad\quad u_{j,i,G} = x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\ \quad \text{else} \\ \quad\quad u_{j,i,G} = x_{j,i,G} \\ \} \end{array}$$

In this DE version, NP must be at least four and it remains fixed along CR and F during the whole execution of the algorithm. Parameter $CR \in [0,1]$, which controls the crossover operation, represents the probability that an element for the trial vector is chosen from a linear combination of three randomly chosen vectors and not from the old vector $\vec{x}_{i,G}$. The condition "$j = j_{rand}$" is to make sure that at least one element is different compared to the elements of the old vector. The parameter F is a scaling factor for mutation and its value is typically $(0, 1+]$[1]. In practice, CR controls the rotational invariance of the search, and its small value (*e.g.*, 0.1) is practicable with separable problems while larger values (*e.g.*, 0.9) are for non-separable problems. The control parameter F controls the speed and robustness of the search, *i.e.*, a lower value for F increases the convergence rate but it also adds the risk of getting stuck into a local optimum. Parameters CR and NP have the same kind of effect on the convergence rate as F has.

[1]Notation means that the upper limit is about 1 but not strictly defined.

0-7803-9487-9/06/$20.00/©2006 IEEE

After the mutation and crossover operations, the trial vector $\vec{u}_{i,G}$ is compared to the old vector $\vec{x}_{i,G}$. If the trial vector has an equal or better objective value, then it replaces the old vector in the next generation. This can be presented as follows (in this paper minimization of objectives is assumed) [4]:

$$\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G} & \text{if} \quad f(\vec{u}_{i,G}) \leq f(\vec{x}_{i,G}) \\ \vec{x}_{i,G} & \text{otherwise} \end{cases} .$$

DE is an elitist method since the best population member is always preserved and the average objective value of the population will never get worse.

III. Generalized Differential Evolution

Generalized Differential Evolution (GDE) extends DE for constrained multi-objective optimization without introducing any extra parameters. There exists different development versions of GDE [5]–[8] but the first version [5] has been used here since the ability for multi-objective optimization is not concerned (different GDE versions differ from the ability for multi-objective optimization). This first GDE version modifies only the selection rule of the basic DE. The basic idea in the selection rule of GDE is that the trial vector is selected to replace the old vector in the next generation if it dominates the old vector weakly [2] in the space of constraint violations or objective functions. Formally this is presented as follows for M objectives f_m and K constraint functions g_k:

$$\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G} & \text{if} \begin{cases} \begin{cases} \exists k \in \{1,\ldots,K\} : \\ \quad g_k(\vec{u}_{i,G}) > 0 \\ \wedge \\ \forall k \in \{1,\ldots,K\} : \\ \quad g'_k(\vec{u}_{i,G}) \leq g'_k(\vec{x}_{i,G}) \end{cases} \\ \vee \\ \begin{cases} \forall k \in \{1,\ldots,K\} : \\ \quad g_k(\vec{u}_{i,G}) \leq 0 \\ \wedge \\ \exists k \in \{1,\ldots,K\} : \\ \quad g_k(\vec{x}_{i,G}) > 0 \end{cases} \\ \vee \\ \begin{cases} \forall k \in \{1,\ldots,K\} : \\ \quad g_k(\vec{u}_{i,G}) \leq 0 \\ \quad \wedge \\ \quad g_k(\vec{x}_{i,G}) \leq 0 \\ \wedge \\ \forall m \in \{1,\ldots,M\} : \\ \quad f_m(\vec{u}_{i,G}) \leq f_m(\vec{x}_{i,G}) \end{cases} \end{cases} \\ \vec{x}_{i,G} & \text{otherwise} \end{cases}$$

, where $g'_k(\vec{x}_{i,G}) = \max(g_k(\vec{x}_{i,G}), 0)$ and $g'_k(\vec{u}_{i,G}) = \max(g_k(\vec{u}_{i,G}), 0)$ are representing the constraint violations. GDE handles any number of M objectives and any number of K constraints, including the cases where $M = 0$ (constraint satisfaction problem) and $K = 0$ (unconstrained problem), and the original DE is a special case when $M = 1$ and $K = 0$. GDE can be implemented in such a way that the number of function evaluations is reduced because not always all the constraints and objectives need to be evaluated, *e.g.*, inspecting constraint violations (even one constraint) is often enough to determine, which vector to select for the next generation [3], [9], [10].

Although GDE uses the most common DE/rand/1/bin strategy, this strategy can be changed to any other DE strategy such as presented in [3], [11]–[13] or, generally, to any method where a child vector is compared against a parent vector and a better one is preserved.

[2] A solution (vector) x weakly dominates solution y in a function space F, if x has at least as good value for all the functions of F as y has. If x weakly dominates y and x has better value for at least one function of F than y has, then x dominates y. If x and y do not dominate each other, they belong to the same non-dominating set.

IV. Experiments

A. Configuration

GDE and given problems were implemented in ANSI-C programming language. Compiler was GCC with an optimization flag -O3. Used hardware was a PC with 2.6 GHz Pentium 4 CPU & 512 MB RAM, and operating system was Linux.

Possible equality constraints $h_j(\vec{x}) = 0$ in the problems were changed to inequality form $|h_j(\vec{x})| - \epsilon \leq 0$, where ϵ was set to 0.0001. In the case of boundary constraints, violating variable values were reflected back from the violated boundary using following rule before the selection operation of GDE:

$$u_{j,i,G} = \begin{cases} 2x_j^{(lo)} - u_{j,i,G} & \text{if} \quad u_{j,i,G} < x_j^{(lo)} \\ 2x_j^{(up)} - u_{j,i,G} & \text{if} \quad u_{j,i,G} > x_j^{(up)} \\ u_{j,i,G} & \text{otherwise} \end{cases} ,$$

where $x_j^{(lo)}$ and $x_j^{(up)}$ are lower and upper limits respectively for a decision variable x_j. If given boundary was not a part of the decision space (*e.g.*, $x_j^{(lo)} < x_j < x_j^{(up)}$), variable at a boundary was moved to inside search bounds by smallest available number (*i.e.*, MINDOUBLE in C).

B. Parameter Setting

Along stopping criteria and size of the population (*NP*) GDE has two control parameters (*CR* and F) as described in Section II, where the effect and ranges of these are also given. As a rule of thumb, if nothing is known about the problem in hand then suitable initial control parameter values are *CR* = 0.9 and $F = 0.9$, and *NP* should be selected from the range $2 \cdot D$ and $40 \cdot D$, where D is number of decision variables of the problem. For an easy problem small *NP* is sufficient but with difficult problems large *NP* is recommended in order to avoid stagnation to a local optimum.

The effect of the parameters on the number of solution candidate evaluations (FES) needed for finding global optimum was studied with the first problem of the problem set defined in [1]. The problem, g01, has 13 decision variables, quadratic objective function, and nine linear inequality constraints. Results are illustrated in Fig. 1, where each curve presents situation when the value of one parameter has been changed while values of others have been kept fixed. One can observe that by increasing parameter values, the number of FES needed also increases. Dependency between NP and number of FES needed is linear while number of FES needed increases more rapidly along CR and F. When $F < 0.2$, search stagnated to a local optimum.

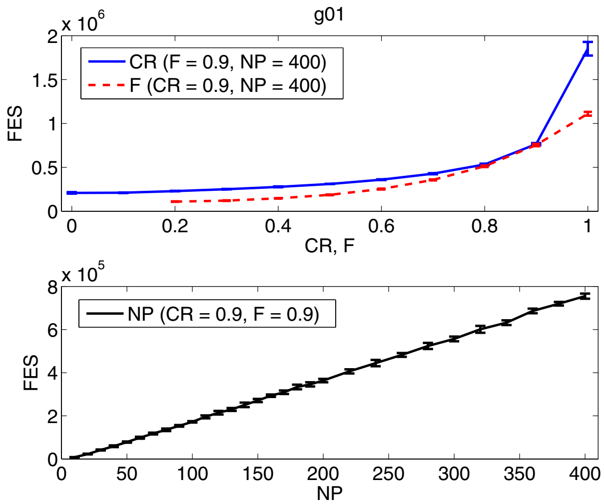

Fig. 1. Effect of control parameters on the number of solutions candidate evaluations (FES) needed for finding the global optimum of g01. Curves represent mean and standard deviation as error bars from 10 runs.

Same control parameter values, $CR = 0.9$, $F = 0.9$, and $NP = 30$, as used in [14] were used also here to solve the problems. The population size smaller than generally recommended was used to speed up the search at the cost of reliability. This was because of defined maximum number of solution candidate evaluations of 500 000 [1], which is relatively small number for finding global optimum reliably for all the problems. In initial tests, when larger population size and number of candidate solution evaluations were used, the global optimum was found for all the problems, except for problem g20.

C. Achieved Results

The 24 problems given in [1] were solved 25 times and history data was recorded during solving process. Achieved results are presented in Tables I–V and Figs. 2–5.

Tables I–IV show the best, the worst, median, mean, and standard deviation after different number of FES for the 25 runs. Numbers in parenthesis after objective function values mark corresponding number of violated constraints. Number of constraints, which are infeasible at a median solution by more than 1, 0.01, and 0.0001, are shown in c as sequences of three numbers respectively. Mean violation [1] for the same median solution is shown in $\overline{v}$. Table V shows number of FES needed for reaching the global optimum, and calculated performance metrics from the runs for the problems.

Figs. 2–5 show development of objective values as well as mean violations of constraints for the median solutions. Convergence data has been collected from GDE by saving values of trial vectors, which have replaced old vectors. Therefore, graphs may have different number of data points and they are not necessarily monotonic. In a way, the graphs tell more about the progress of the search than graphs, which only show values of the best individual in each generation. Also, one should note that because of logarithmic scale of horizontal axes, values which are negative or zero have not been shown. Quite often the error of an objective function value is less than zero before finding a feasible solution and it becomes zero when a global optimum has been reached.

As it can be seen from Tables I–IV, GDE was able to find the feasible global optimum with 0.0001 accuracy for almost all the problems in 500 000 solution candidate evaluations. According to Success Performance in Table V, difficulty of problems increases in following order: g08, g24, g12, g06, g11, g16, g04, g09, g01, g15, g10, g07, g02, g05, g14, g19, g18, g21, g13, g23, g17, g03, g20, and g22. Problems g03, g20, and g22 turned out to be most difficult ones. Feasible global optimum for g03 was found only once and more often search had converged to a local optimum as it can be seen from Fig. 2. Problem g20 is probably the most difficult one among the problems; a feasible solution was never found and this problem is thought to be infeasible. When g20 was tested before performing actual tests for the problems, constraints of g20 were found to conflict with each others. This was tested by creating a large set of decision vectors randomly and then testing non-dominance of corresponding constraint violations. It was found that more than 99% of these were non-dominating. In GDE an old infeasible solution is replaced with a new solution only if the new solution dominates the old one weakly. This property together with the conflicting constraint set of g20 caused that there was no improvement during search process and therefore there is no convergence graph for this problem in Fig. 5. There was only a little improvement with g22 and a feasible solution was never found in 500 000 solution candidate evaluations. However, also this problem was solvable with larger population and number of FES.

A feasible solution better than the optimum given in [1] was found for three problems (marked with boldface in Tables I–IV). Two out of these contain equality constraints and an improvement is meaningless due to the allowed error in the constraint evaluations. For one the new solution is an improvement in the sense of presentation accuracy. Details of these solutions are given in Appendix.

One should note that GDE has used a lower number of objective and constraint function evaluations than the majority of the other optimization methods, where typically evaluation of a solution candidate means evaluation of the objective function and all the constraint functions involved. GDE needs a considerably lower number of function evaluations because use of dominance principle in selection explained in Section III. Often the selection can be performed without evaluating all the objective and constraint functions involved. This is demonstrated in Table VI, where actual number of function evaluations has been reported for one run of the

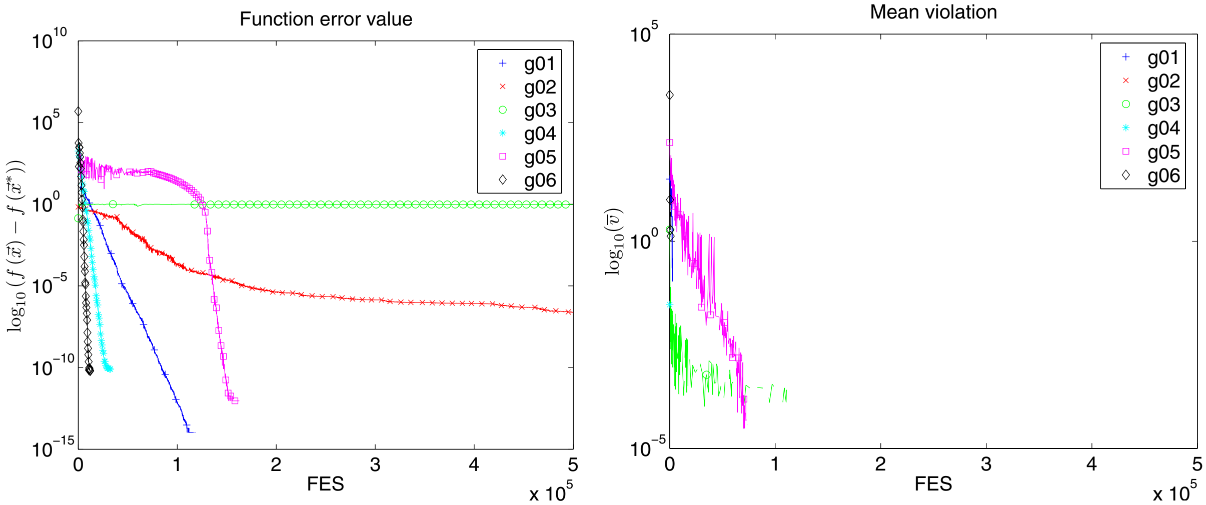

Fig. 2. Convergence graph for problems 1–6.

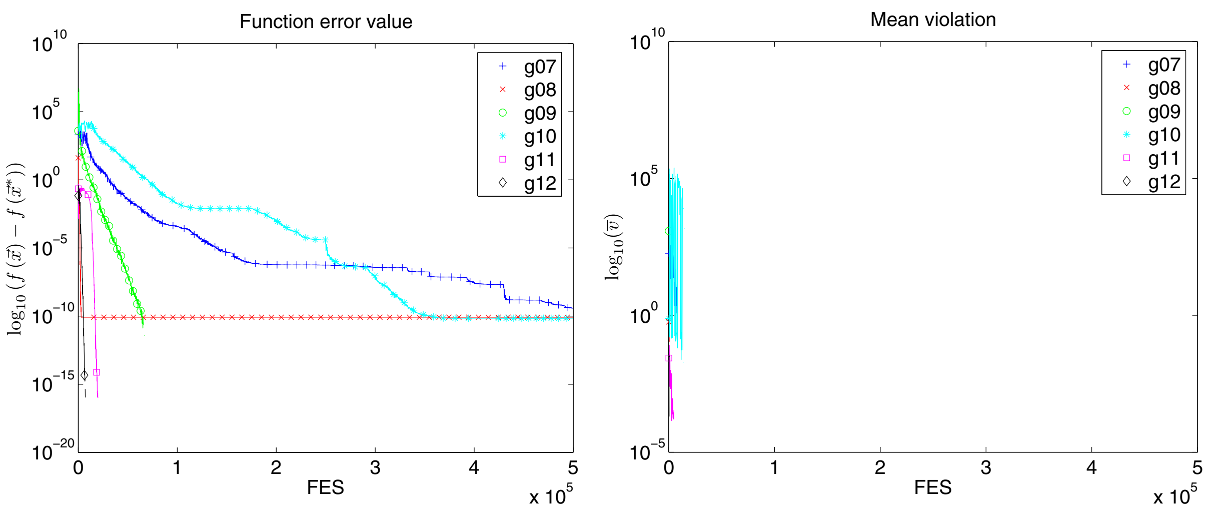

Fig. 3. Convergence graph for problems 7–12.

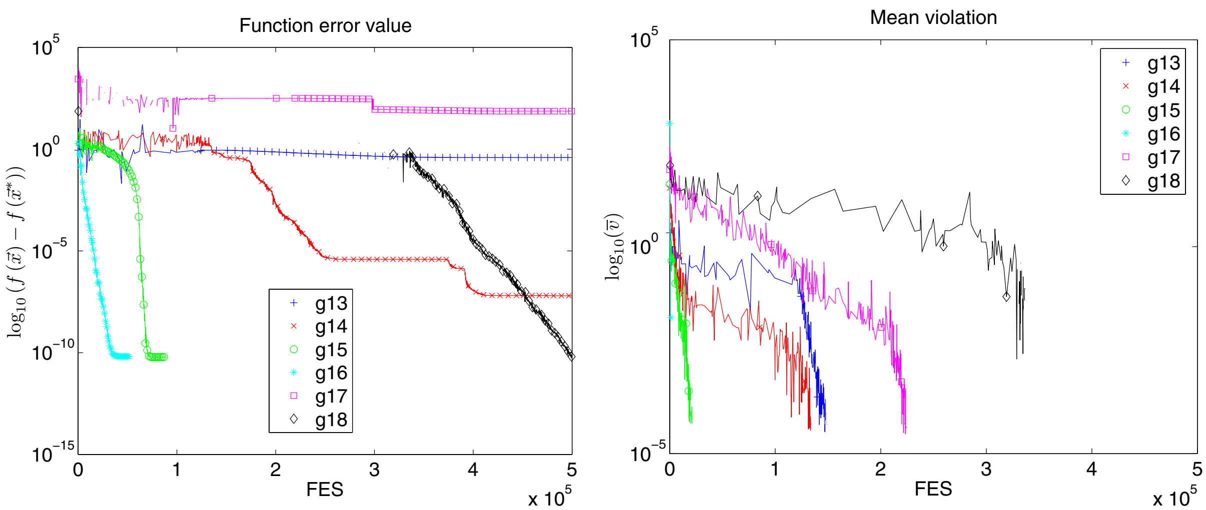

Fig. 4. Convergence graph for problems 13–18.

TABLE I

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 1–6

FES		g01	g02	g03	g04	g05	g06
	Best	3.5763e+00 (0)	2.5154e-01 (0)	9.1747e-01 (0)	1.2570e+00 (0)	6.8587e+01 (3)	4.1743e-04 (0)
	Median	4.7424e+00 (0)	4.2748e-01 (0)	9.6146e-01 (1)	4.2108e+00 (0)	4.4669e+01 (3)	5.8119e-03 (0)
	Worst	6.1002e+00 (0)	5.3733e-01 (0)	9.9822e-01 (1)	9.6515e+00 (0)	1.1013e+02 (3)	2.6624e-02 (0)
$\mathbf{5\times10^3}$	c	0,0,0	0,0,0	0,0,0	0,0,0	3,3,3	0,0,0
	$\overline{v}$	0.0000e+00	0.0000e+00	1.0392e-04	0.0000e+00	5.5238e+00	0.0000e+00
	Mean	4.8746e+00	4.2004e-01	9.5737e-01	4.2753e+00	2.0198e+02	9.1263e-03
	Std	7.0208e-01	5.5779e-02	6.4713e-02	1.9422e+00	2.8829e+02	8.0013e-03
	Best	1.4320e-06 (0)	3.7882e-03 (0)	5.6891e-01 (0)	8.0036e-11 (0)	1.4551e+00 (0)	6.1846e-11 (0)
	Median	4.6499e-06 (0)	2.3988e-02 (0)	9.5472e-01 (0)	8.0036e-11 (0)	6.0133e+01 (3)	6.1846e-11 (0)
	Worst	1.5465e-05 (0)	5.0705e-02 (0)	9.9739e-01 (1)	8.0036e-11 (0)	2.8090e+00 (3)	6.1846e-11 (0)
$\mathbf{5\times10^4}$	c	0,0,0	0,0,0	0,0,0	0,0,0	0,2,3	0,0,0
	$\overline{v}$	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	5.4566e-03	0.0000e+00
	Mean	5.1730e-06	2.5299e-02	9.2391e-01	8.0036e-11	3.1510e+02	6.1846e-11
	Std	3.1523e-06	1.2887e-02	1.0059e-01	0.0000e+00	2.8043e+02	0.0000e+00
	Best	0.0000e+00 (0)	5.0812e-08 (0)	**-9.8170e-12 (0)**	8.0036e-11 (0)	0.0000e+00 (0)	6.1846e-11 (0)
	Median	0.0000e+00 (0)	2.3251e-07 (0)	9.3634e-01 (0)	8.0036e-11 (0)	0.0000e+00 (0)	6.1846e-11 (0)
	Worst	0.0000e+00 (0)	2.2776e-02 (0)	9.9739e-01 (1)	8.0036e-11 (0)	8.2996e+02 (2)	6.1846e-11 (0)
$\mathbf{5\times10^5}$	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	Mean	0.0000e+00	4.3170e-03	8.9606e-01	8.0036e-11	4.0583e+01	6.1846e-11
	Std	0.0000e+00	7.5112e-03	1.9833e-01	0.0000e+00	1.6854e+02	0.0000e+00

TABLE II

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 7–12

FES		g07	g08	g09	g10	g11	g12
	Best	1.5353e+02 (0)	8.1964e-11 (0)	4.8941e+00 (0)	7.8166e+03 (0)	3.1281e-09 (0)	1.3510e-12 (0)
	Median	2.5982e+03 (1)	8.1964e-11 (0)	1.7976e+01 (0)	2.2104e+04 (1)	3.8076e-02 (0)	5.8919e-11 (0)
	Worst	2.3567e+03 (4)	8.1964e-11 (0)	4.4106e+01 (0)	1.6205e+04 (2)	1.8728e-01 (0)	1.4535e-07 (0)
$\mathbf{5\times10^3}$	c	1,1,1	0,0,0	0,0,0	0,1,1	0,0,0	0,0,0
	$\overline{v}$	4.9051e-01	0.0000e+00	0.0000e+00	5.9098e-03	0.0000e+00	0.0000e+00
	Mean	1.3836e+03	8.1964e-11	1.8793e+01	1.2567e+04	4.9668e-02	1.1825e-08
	Std	9.4010e+02	0.0000e+00	9.1478e+00	3.7524e+03	5.4531e-02	3.7439e-08
	Best	1.6983e-02 (0)	8.1964e-11 (0)	3.4845e-09 (0)	1.2578e+00 (0)	0.0000e+00 (0)	0.0000e+00 (0)
	Median	3.8955e-02 (0)	8.1964e-11 (0)	2.7078e-08 (0)	6.1670e+00 (0)	0.0000e+00 (0)	0.0000e+00 (0)
	Worst	1.4710e-01 (0)	8.1964e-11 (0)	1.7606e-07 (0)	2.9779e+01 (0)	0.0000e+00 (0)	0.0000e+00 (0)
$\mathbf{5\times10^4}$	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	Mean	5.5401e-02	8.1964e-11	4.1921e-08	8.7481e+00	0.0000e+00	0.0000e+00
	Std	4.0648e-02	0.0000e+00	4.1657e-08	6.9854e+00	0.0000e+00	0.0000e+00
	Best	7.9822e-11 (0)	8.1964e-11 (0)	**-9.7884e-11 (0)**	6.9122e-11 (0)	0.0000e+00 (0)	0.0000e+00 (0)
	Median	3.6402e-10 (0)	8.1964e-11 (0)	-9.7884e-11 (0)	6.9122e-11 (0)	0.0000e+00 (0)	0.0000e+00 (0)
	Worst	1.6532e-06 (0)	8.1964e-11 (0)	-9.7771e-11 (0)	6.9122e-11 (0)	0.0000e+00 (0)	0.0000e+00 (0)
$\mathbf{5\times10^5}$	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	Mean	1.2966e-07	8.1964e-11	-9.7884e-11	6.9122e-11	0.0000e+00	0.0000e+00
	Std	3.8948e-07	0.0000e+00	4.9555e-14	0.0000e+00	0.0000e+00	0.0000e+00

problem g17. Also, total number of function evaluations compared to evaluating all the functions for every solution candidate (AFES) has been shown as percentile. In the last column of Table V the same percentile has been presented for all the problems and runs. For example, with the problems g20 and g22 the number of actual function evaluations is only about 10% from the maximum number of evaluations and therefore it would had been more fair to have ten times more solution candidate evaluations for these problems when results are compared with an optimization method, which evaluates all the functions for all the solution candidates.

Another thing to note is that rearranging constraint functions would have affected the actual number of function evaluations as well as the execution time. Evaluating first the constraints, which are harder to satisfy (*e.g.*, equality constraint and non-linear functions) would have decreased the number of other function evaluations. Also, evaluating computationally expensive constraint functions (in this case those containing mathematical functions, *e.g.*, `sin`, `cos`, `log`, `sqrt`, and `pow`) last would have decreased the execu-

TABLE III

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 13–18

FES		g13	g14	g15	g16	g17	g18
$\mathbf{5 \times 10^3}$	Best	1.4609e+00 (3)	7.9473e+00 (3)	2.9401e-03 (2)	6.5114e-03 (0)	3.4733e+02 (4)	2.6320e+00 (9)
	Median	6.9500e-01 (3)	-1.0704e+01 (3)	9.3370e+00 (2)	1.4218e-02 (0)	-1.0769e+02 (4)	-2.4075e+01 (10)
	Worst	3.0518e+00 (3)	-8.9451e+01 (3)	3.6083e+00 (2)	4.0254e-02 (0)	-2.1707e+02 (4)	-8.8261e+00 (9)
	c	0,3,3	1,3,3	0,1,2	0,0,0	4,4,4	10,10,10
	$\overline{v}$	2.4416e-01	5.4482e-01	2.7754e-02	0.0000e+00	1.4729e+01	2.2040e+01
	Mean	1.9795e+00	-2.2653e+01	2.7388e+00	1.7637e-02	1.3326e+02	-6.4223e+00
	Std	3.0704e+00	2.6434e+01	2.7242e+00	8.5517e-03	9.2486e+02	1.2285e+01
$\mathbf{5 \times 10^4}$	Best	9.4492e-01 (2)	6.6203e+00 (1)	6.0936e-11 (0)	6.5216e-11 (0)	9.0045e+01 (4)	-9.6282e-01 (7)
	Median	9.3720e-01 (3)	6.2038e+00 (3)	4.7556e-01 (0)	6.5218e-11 (0)	2.3074e+02 (4)	-3.8480e+00 (8)
	Worst	9.4025e-01 (3)	1.6376e+00 (3)	4.8047e+00 (0)	6.5254e-11 (0)	6.1422e+01 (4)	-1.8270e+01 (9)
	c	0,2,3	0,0,3	0,0,0	0,0,0	3,4,4	7,8,8
	$\overline{v}$	6.6307e-02	2.6616e-03	0.0000e+00	0.0000e+00	2.1602e+00	1.0711e+01
	Mean	9.0992e-01	5.0752e+00	1.0998e+00	6.5220e-11	1.6172e+02	-4.9981e+00
	Std	4.5313e-01	1.7452e+00	1.4669e+00	7.5951e-15	1.3549e+02	5.6024e+00
$\mathbf{5 \times 10^5}$	Best	4.1898e-11 (0)	9.1518e-12 (0)	6.0936e-11 (0)	6.5216e-11 (0)	**-5.8000e-03 (0)**	1.5561e-11 (0)
	Median	3.8486e-01 (0)	6.3148e-08 (0)	6.0936e-11 (0)	6.5216e-11 (0)	7.4052e+01 (0)	4.6362e-11 (0)
	Worst	1.4403e+00 (3)	1.9281e-02 (0)	4.8047e+00 (0)	6.5217e-11 (0)	2.8637e+02 (4)	-3.1844e+00 (6)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	Mean	3.4383e-01	7.7373e-04	1.9219e-01	6.5216e-11	8.6678e+01	-1.6271e-01
	Std	3.8650e-01	3.8556e-03	9.6094e-01	1.7764e-16	8.2114e+01	6.4619e-01

TABLE IV

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 19–24

FES		g19	g20	g21	g22	g23	g24
$\mathbf{5 \times 10^3}$	Best	6.2493e+01 (0)	1.4891e+01 (20)	5.7434e+02 (5)	5.6350e+03 (19)	3.6407e+02 (4)	4.9104e-09 (0)
	Median	1.2402e+02 (0)	1.3503e+01 (20)	8.5047e+01 (6)	8.1251e+03 (19)	1.0173e+02 (5)	2.6790e-08 (0)
	Worst	1.9009e+02 (0)	1.4233e+01 (20)	8.0060e+02 (5)	1.5154e+04 (19)	8.2694e+02 (6)	1.4833e-07 (0)
	c	0,0,0	2,15,20	2,6,6	18,19,19	3,5,5	0,0,0
	$\overline{v}$	0.0000e+00	8.6860e+00	1.2002e+01	2.9497e+08	3.7091e+00	0.0000e+00
	Mean	1.3283e+02	1.4977e+01	4.5058e+02	1.0896e+04	1.5217e+02	3.9665e-08
	Std	3.6448e+01	1.9580e+00	1.9713e+02	6.2602e+03	4.4607e+02	3.2954e-08
$\mathbf{5 \times 10^4}$	Best	4.0838e-01 (0)	1.4891e+01 (20)	4.6430e+02 (5)	5.6350e+03 (19)	3.3928e+02 (4)	4.7269e-12 (0)
	Median	1.0853e+00 (0)	1.3503e+01 (20)	5.5839e+02 (5)	9.5962e+03 (19)	5.0940e+02 (4)	4.7269e-12 (0)
	Worst	2.4155e+00 (0)	1.4233e+01 (20)	6.3921e+02 (5)	1.5154e+04 (19)	3.1369e+02 (4)	4.7269e-12 (0)
	c	0,0,0	2,15,20	1,4,5	18,19,19	1,3,4	0,0,0
	$\overline{v}$	0.0000e+00	8.6860e+00	2.0259e+00	2.8489e+08	5.3501e-01	0.0000e+00
	Mean	1.2173e+00	1.4977e+01	4.6012e+02	1.0080e+04	4.4875e+02	4.7269e-12
	Std	5.6659e-01	1.9580e+00	2.3014e+02	6.0804e+03	2.4058e+02	0.0000e+00
$\mathbf{5 \times 10^5}$	Best	4.6448e-11 (0)	1.4891e+01 (20)	3.4987e-11 (0)	1.9654e+04 (19)	1.3112e-08 (0)	4.7269e-12 (0)
	Median	5.2669e-09 (0)	1.3503e+01 (20)	6.5523e-08 (0)	9.7885e+03 (19)	1.0569e+01 (0)	4.7269e-12 (0)
	Worst	1.8287e-04 (0)	1.4233e+01 (20)	2.3612e+02 (5)	1.5154e+04 (19)	6.4487e+02 (4)	4.7269e-12 (0)
	c	0,0,0	2,15,20	0,0,0	16,19,19	0,0,0	0,0,0
	$\overline{v}$	0.0000e+00	8.6860e+00	0.0000e+00	2.4366e+08	0.0000e+00	0.0000e+00
	Mean	1.7894e-05	1.4977e+01	6.3186e+01	1.0647e+04	9.2414e+01	4.7269e-12
	Std	4.5735e-05	1.9580e+00	8.6788e+01	5.9865e+03	1.9119e+02	0.0000e+00

tion time, since later the constraint is in evaluation order, the fewer times it gets evaluated. Order of constraint functions have also an effect on search process since search is directed at the beginning according to first constraints. This property may be both useful and harmful. However, rearrangements of constraints was not done to keep things simple and clear.

D. Algorithm Complexity

The GDE version is linearly scalable algorithm with simple genetic operations of DE. Therefore, also problems with large number of decision variables and/or large population size are solvable in reasonable time.

The complexity of GDE was measured according to instructions given in [1], and observed CPU times are reported in Table VII.

V. CONCLUSIONS

Results with Generalized Differential Evolution (GDE) for the CEC 2006 Special Session on Constrained Real-Parameter Optimization have been shown. According to

TABLE V

Number of FES to achieve the fixed accuracy level $((f(\vec{x}) - f(\vec{x}^*)) \leq 0.0001)$, Success Rate, Feasible Rate, and Success Performance [1]. Also, percentage of actual number of function evaluations compared to maximum amount of function evaluations (AFES)

Prob.	Best	Median	Worst	Mean	Std	Feasible Rate	Success Rate	Success Perf.	AFES
g01	38076	40200	43838	40519	1652.2000	100%	100%	40519	96.2%
g02	93550	106332	124386	107684	10947.4011	100%	72%	149561	66.9%
g03	143086	143086	143086	143086	0.0000	96%	4%	3577150	93.4%
g04	13679	15157	17692	15281	880.5684	100%	100%	15281	92.2%
g05	61057	186461	298724	178023	57263.7453	96%	92%	193503	81.1%
g06	6101	6431	7160	6503	292.6923	100%	100%	6503	96.2%
g07	87437	112969	412908	123996	62469.0586	100%	100%	123996	47.5%
g08	1178	1486	1822	1469	159.8284	100%	100%	1469	99.9%
g09	25743	30784	33140	30230	1873.2359	100%	100%	30230	67.6%
g10	67344	81827	101487	82604	7548.5327	100%	100%	82604	65.5%
g11	1639	8753	14523	8460	3853.2721	100%	100%	8460	77.4%
g12	2419	3016	4422	3149	453.3189	100%	100%	3149	99.7%
g13	212723	332054	474184	336306	90225.9255	88%	40%	840766	53.2%
g14	157455	200668	365537	220921	56917.8358	100%	96%	230126	55.9%
g15	20089	74038	122399	71889	27631.2386	100%	96%	74885	67.8%
g16	11001	13307	15296	13224	965.7453	100%	100%	13224	93.2%
g17	215578	330592	498198	343740	139769.5095	76%	16%	2148377	31.3%
g18	169424	377732	499909	364861	89960.0295	84%	76%	480080	19.5%
g19	162297	206556	244114	202648	23418.8023	100%	88%	230282	45.2%
g20	–	–	–	–	–	0%	0%	–	9.1%
g21	176939	347881	493247	347653	88560.1500	88%	60%	579422	40.0%
g22	–	–	–	–	–	0%	0%	–	11.7%
g23	341545	437256	461521	425342	37808.5232	88%	40%	1063354	41.6%
g24	2656	3059	3408	3059	191.5507	100%	100%	3059	88.5%

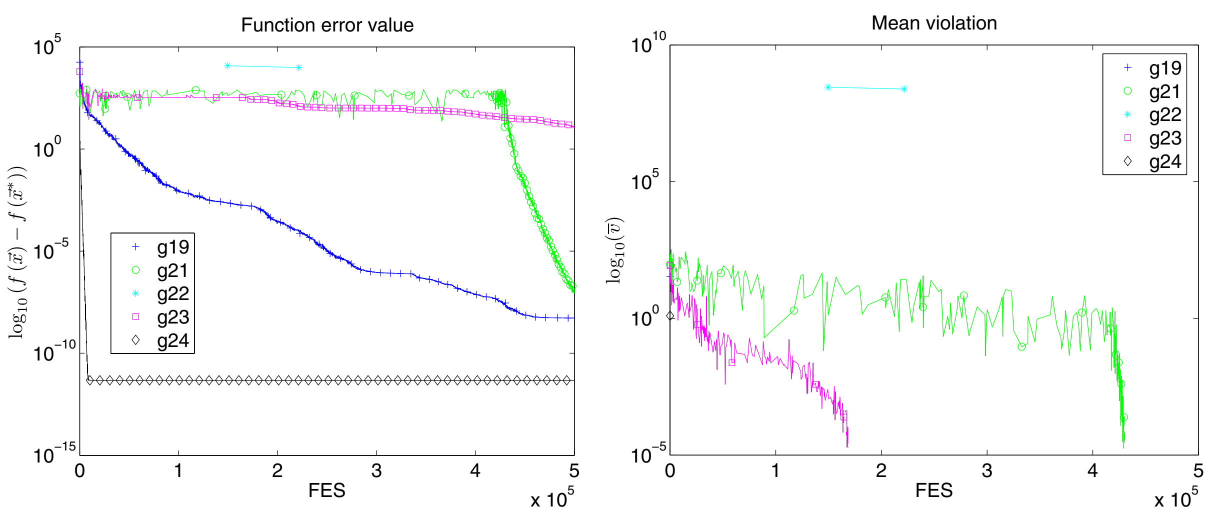

Fig. 5. Convergence graph for problems 19–24.

them, GDE was able to solve almost all the problems in the given maximum number of solution candidate evaluations but not with total reliability.

The problems were solved using those control parameter values, which are generally recommended to be applied as first guess settings for a new problem. The size of the population was set smaller than recommended, since the hardly limited maximum number of solution candidate evaluations required to favor high convergence speed more than usually. This probably decreased reliability of finding the global optimum of the problems. One should note especially that all the test problems were solved with the same control parameter settings, and no kind of problem specific tuning was performed.

The constraint handling method of GDE decreases the actual number of needed objective and constraint function evaluations, and therefore more generations or solution candidate evaluations could be performed in the time taken by

TABLE VI

ACTUAL NUMBER OF CONSTRAINT AND OBJECTIVE FUNCTION EVALUATIONS FOR THE G17 PROBLEM, AND PERCENTAGE OF ACTUAL NUMBER OF FUNCTION EVALUATIONS COMPARED TO MAXIMUM AMOUNT OF FUNCTION EVALUATIONS

h_1	h_2	h_3	h_4	f	AFES
500000	215003	131309	89758	72742	33.6%

TABLE VII

COMPUTATIONAL COMPLEXITY ACCORDING TO [1]: T1 IS MEAN TIME FOR EVALUATING ALL THE PROBLEMS (*i.e.*, EVALUATING THE OBJECTIVE AND ALL THE CONSTRAINTS OF THE PROBLEMS), AND T2 IS MEAN TIME FOR SOLVING ALL THE PROBLEMS USING GDE

$T1$	$T2$	$(T2-T1)/T1$
0.006771 s	0.020479 s	2.0245

an optimization technique, which evaluates all the functions for all the solution candidates.

APPENDIX

Here are presented details of obtained best solutions when a new solution was considered better than in [1]. Problems g03 and g17 contain equality constraints and the improvement is probably because of allowed error ϵ in equality function evaluations. Problem g09 contains inequality constraints and its solution is improved in the sense of presentation accuracy.

Vector $\vec{x}*$ shows decision variables and $f(\vec{x}*)$ is corresponding objective function value with 17 decimals accuracy[3].

g03:

$\vec{x}*$ =
(3.16243578702882222e-01, 3.16243576699840712e-01, 3.16243577448651170e-01, 3.16243577009048371e-01, 3.16243575328125415e-01, 3.16243576139409843e-01, 3.16243578113241774e-01, 3.16243576723850672e-01, 3.16243577546733934e-01, 3.16243576386954661e-01)
and $f(\vec{x}*)$ = 1.00050010001000222e+00.

g09:

$\vec{x}*$ =
(2.33049942166640145e+00, 1.95137237414051956e+00, -4.77541394531771246e-01, 4.36572621642362257e+00, -6.24486962864802209e-01, 1.03813093086817720e+00, 1.59422672847830094e+00)
and $f(\vec{x}*)$ = 6.80630057374402213e+02.

g17:

$\vec{x}*$ =
(2.01784462493549540e+02, 9.99999999999999432e+01, 3.83071034852772868e+02, 4.19999999999999659e+02, -1.09077879765636379e+01, 7.31482312084287961e-02)
and $f(\vec{x}*)$ = 8.85353387480648416e+03.

[3]Number of digits is slightly more than reliable machine accuracy and therefore a couple of last decimals might be wrong.

ACKNOWLEDGMENTS

First author gratefully acknowledges support from the East Finland Graduate School in Computer Science and Engineering (ECSE), Technological Foundation of Finland, Finnish Cultural Foundation, and Centre for International Mobility (CIMO). He also wishes to thank all the KanGAL people for their help and fruitful discussions.

REFERENCES

[1] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb, “Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization,” School of EEE, Nanyang Technological University, Singapore, 639798, Technical Report, December 2005, [Online] Available: http://www.ntu.edu.sg/home/EPNSugan, 14.4.2006.

[2] R. Storn and K. V. Price, “Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec 1997.

[3] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag, 2005.

[4] K. V. Price, *New Ideas in Optimization*. London: McGraw-Hill, 1999, ch. An Introduction to Differential Evolution, pp. 79–108.

[5] J. Lampinen, “DE’s selection rule for multiobjective optimization,” Lappeenranta University of Technology, Department of Information Technology, Tech. Rep., 2001, [Online] Available: http://www.it.lut.fi/kurssit/03-04/010778000/MODE.pdf, 15.2.2006.

[6] S. Kukkonen and J. Lampinen, “Comparison of Generalized Differential Evolution algorithm to other multi-objective evolutionary algorithms,” in *Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS2004)*, Jyväskylä, Finland, July 2004, p. 445.

[7] ——, “An extension of Generalized Differential Evolution for multi-objective optimization with constraints,” in *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Birmingham, England, Sept 2004, pp. 752–761.

[8] ——, “GDE3: The third evolution step of Generalized Differential Evolution,” in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, Scotland, Sept 2005, pp. 443–450.

[9] J. Lampinen, “Multi-constrained nonlinear optimization by the Differential Evolution algorithm,” Lappeenranta University of Technology, Department of Information Technology, Tech. Rep., 2001, [Online] Available: http://www.it.lut.fi/kurssit/03-04/010778000/DECONSTR.PDF, 15.2.2006.

[10] ——, “A constraint handling approach for the Differential Evolution algorithm,” in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Honolulu, Hawaii, May 2002, pp. 1468–1473.

[11] D. Zaharie, “Multi-objective optimization with adaptive Pareto Differential Evolution,” in *Proceedings of Symposium on Intelligent Systems and Applications (SIA 2003)*, Iasi, Romania, Sept 2003.

[12] V. Feoktistov and S. Janaqi, “New strategies in Differential Evolution,” in *Proceedings of the 6th International Conference on Adaptive Computing in Design and Manufacture (ACDM2004)*, Bristol, United Kingdom, April 2004, pp. 335–346.

[13] V. L. Huang, A. K. Qin, and P. N. Suganthan, “Self-adaptive Differential Evolution algorithm for constrained real-parameter optimization,” in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 2006, accepted for publication.

[14] J. Rönkkönen, S. Kukkonen, and K. V. Price, “Real-parameter optimization with differential evolution,” in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, Scotland, Sept 2005, pp. 506–513.